Best
Available
Copy

AD-772 828

ANALYTIC SOLUTION OF COUPLED MODE EQUATIONS BY COMPUTER

Bernard Rosen

Stevens Institute of Technology

Prepared for:

Army Research Office-Durham
Advanced Research Projects Agency

7 December 1973

DISTRIBUTED BY:

NTIS
National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

Final Technical Report

Analytic Solution of Coupled Mode Equations by Computer

Bernard Rosen

December 7, 1973

Approved for public release;

distribution unlimited

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by U.S. Army Research Office-Durham under Grant/Contract No. DA-ARO-D-31-124-72-G73.

Form Approved
Budget Bureau
No. 22-R0293

ARPA Order No. 1992

Grant No. DA-ARO-D-31-124-72-G73

Project Code No. G2706D

Principal Investigator-
Prof. Bernard Rosen

Name of Grantee-
Stevens Institute of Technology

Effective Date of Grant-
12/20/71

Title-
Final Technical Report - Analytic Solution of Coupled Mode Equations by Computer.

Expiration Date - 12/19/72
Extended to - 3/31/73

Date- January 2, 1974

Amount of Grant - $38,426

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 1992

Approved for Public Release;
distribution unlimited

I. INTRODUCTION

We are using automated symbolic manipulation to generate approximate solutions to the prognostic equations of meteorology. These equations are treated in the form that would arise by means of modal analysis and truncation. Consequently the equations take the form of coupled non-linear first-order ordinary differential equations; the number of such equations may be very large if many modes are included in the analysis. We have been principally concerned with the formulation of a general method in the work reported here.

This report officially covers the period from Dec. 20, 1971 through March 31, 1973. Some of the work reported here reached its natural conclusion in the months following and was influenced by comments made after a talk on these matters by the author at the Rand Corporation in March 1973 at an ARPA meeting on climatology. In the course of this first fifteen months we have made some false starts as we attempted to orient ourselves in this work. We do not report on these here. This is the sole written report on this work; work along these lines continues under ARPA Grant # DA-ARO-D-31-124.

This work has been performed in conjunction with an extensive program to investigate climate prediction and modification and consequently we are interested in the long term behavior of the atmosphere under the influence of the driving 'force' of the sun. The existence of the yearly cycle of seasons would indicate that the atmosphere is acting as a steady state system under the influence of the yearly heating cycle. We might then envisage that climate modification will occur if the heating effect on the atmosphere shows a long term change; naturally if we also took into account

the hydrological cycle other forms of climate modification could be anticipated.

The principal accomplishments during the period of this grant have been

1. Programs useful for the analytic solution of coupled mode equations were developed and tested. These programs were written in PL1/FORMAC.[1] Some have been translated into MACSYMA.[2]

2. Several analytic methods were investigated for their possible bearing upon the problem.

During the term of this grant the principal investigator was the only senior participant in this work; two graduate students, Michael Polcari (Feb. '72 - Jan '73) and Theodore J. Roycraft (Feb. '73 -) were supported by this grant.

The mathematical problem that we face is the solution of coupled non-linear equations containing explicit driving terms whose diurnal frequency is large compared to that of natural oscillations of the driven system. The envelope of the amplitude of the driving terms have a slow time seasonal oscillation and perhaps a longer time scale variation because of variations in the sun's output. In particular the natural modes, i.e. the cyclones and anticyclones of the middle latitude disturbances, have frequencies corresponding to the period of about a week. Inasmuch as the large scale turbulence accounts for the bulk of the energy transport to the poles we cannot ignore the presence of the high frequency modes in the solution of the differential equations. As is well known it is possible because of the non-linear nature of the physics and the equations that high frequency terms can have a low frequency effect. Before we can attempt to solve the steady state problem we

must in some way account for effects of the high frequency modes - that is, the baroclinic instabilities. Barring our ability to solve the equations exactly the average effects of the high frequency modes must be calculated by some approximation technique. One basic difficulty is that the saturation level of the excitation of these natural modes is not given. In the approach taken here we will replace the non-linear effects of the oscillating modes by constant and linear terms in the differential equation. This approach is closely related to the introduction of 'eddy' viscosity and of course the physical ideas are similar.

There are two interconnected problems that are of interest:

a) study of instabilities to determine the level of saturation;

b) study of steady state behavior under driving forces.

The values of the dynamic variables about which we linearize the equations is not chosen arbitrarily to be zero but rather by inspection of the results of numerical integration of the equations or by inspection of the equations themselves.

The analytical work to be done by computer is hedged by a number of inherent restrictions beyond the obvious ones that the results equal or approximate the correct solution. Among these restrictions are the following:

a) It must be possible to specify the steps to be performed in algorithmic fashion so that the procedure can be mechanized.

b) The number of terms that are generated should not be so large that no conclusion could be drawn from an inspection of the analytical results.

c) The calculation should not be a numerical evaluation in disguise.

The results of a high order perturbation-theoretic calculation, for example,

involving many coupled modes will not satisfy conditions (b) and (c). We expect that a combination of numerical and analytic methods may prove most useful; the numerical evaluation of various analytic terms will serve to delineate those parts of the multi-termed analytic answer which are most important.

The method we employ is a variant of the well-known Bogoliubov-Krylov-Mitropolsky (BKM) scheme.[3] This method is used to generate the so-called slow-time-scale equations for the amplitude and the frequency of a non-linear oscillation. The starting point of the entire procedure that we often follow is numerical calculations from which we can determine a reasonable beginning approximation.

In the next section we shall discuss the mathematics of the method we employ and its implementation to date. In Section III we discuss a model for the saturation of the baroclinic instability which illustrates some of the ideas worked out in the previous section. In Section IV we discuss the application to a system of three coupled modes whose free oscillations are described by elliptic functions and some preliminary results on the theory of rotating basin phenomena as formulated by E. N. Lorenz.[3]

II. Mathematics and Implementation

A. Mathematics

Our basic goal is to use the information obtained by linearization of the equations and by perturbative expansion of the solutions to determine a replacement for the non-linear terms in the differential equations in the form of constants and of terms linear in the mode amplitudes. This is the gist of the BKM method of equivalent linearization. Once this step has been made the resulting equations should be quite easy to solve by use of standard techniques used in the analysis of linear systems. The equations that we consider are of the form

$$\frac{dB_i}{dt} = \sum_j M_{ij} B_j + \sum_{jk} \Gamma_{jk}{}^i B_k B_j + D_i(t) \qquad \text{(II-1)}$$

where the B_i are the mode amplitudes, the M_{ij} and the Γ_{ijk} are constants, and the D_i are the driving terms which are explicit functions of time.

In general we will be looking for periodic solutions of the coupled mode equations III-1 by perturbational analysis. Often the results of linearization will lead to growing or decaying solutions rather than oscillating ones. In such cases we may have to change the values of the B_i about which one linearizes. (For an alternate approach see the next section, (Eq. III-6)). For example let us consider one of the most famous nonlinear coupled mode equations, the so called predator-prey equations.[4] Let x and y denote the values of two populations, namely a prey population x and a predator population y. The prey x will increase exponentially

except that the predators eat it up at a rate proportional to the produce of the two populations.

In symbols

$$\frac{dx}{dt} = \alpha x - \beta yx$$

Similarly the predators would decrease exponentially if there were no prey but increases in proportion to the number of prey encountered. So

$$\frac{dy}{dt} = - \gamma y + \delta xy$$

we can make the identifications $[M_{11} = \alpha\ ,\ M_{22} = -\gamma\ ,\ \Gamma_{12}{}^{1} = -\beta\ ,\ \Gamma_{21}{}^{2} = \delta]$

If we were to linearize these equations about the point of equilibrium $x = y = 0$, we would find that the prey increases as exp-At and the predators decrease as exp-Bt. Carrying out perturbation theory will not lead us away from real exponentials whereas the actual populations oscillate. We can obtain this behavior by linearizing about a second point of equilibrium $y_o = \alpha/\beta$, $X_o = \gamma/\delta$. Writing $Y = \beta y - \alpha$, $X = \delta x - \gamma$, we get

$$\frac{dX}{dt} = - \gamma Y - XY$$

$$\frac{dY}{dt} = \alpha Y + XY$$

If one linearizes this pair of equations about the point $X = Y = 0$ the resulting set has an oscillatory solution at the frequency $2\pi f = \sqrt{\alpha\gamma}$

In any event, based upon the results of the numerical integration and/or some analytical study of the set of Equations III-1 we introduce mode amplitudes u_i by $B_i = u_i + B_{io}$ we then have:

$$\frac{du_i}{dt} = \sum_j M_{ij}^1 u_j + \varepsilon \sum_{j,k} \Gamma_{jk}{}^i u_j u_k + D_i{}^1 \qquad \text{(II-2)}$$

where:

$$M_{ij}{}^1 = M_{ij} + \sum_k (\Gamma_{ijk}{}^i B_{k,o} + \Gamma_{k,j}{}^i B_{k,o})$$

$$D_i{}^1 = D_i + \sum_j M_{ij} B_{jo} + \sum_{jk} \Gamma_{jk}{}^i B_{jo} B_{ko}$$

We have introduced a purely formal ordering parameter ε which is considered small for the purposes of carrying out a perturbation expansion but which in actuality is equal to 1.

In order to make the analysis seem less abstract we carry out the steps indicated symbolically on a specific example, viz. a driven harmonic oscillator on which a fictional force of the form $\varepsilon v|v|$ acts (v is the velocity). The equations governing the notion of this oscillator are

$$\frac{dx}{dt} = v$$

$$\frac{dv}{dt} = - \omega^2 x - \varepsilon v|v| + D \sin\Omega t$$

The next step is to find the eigenmodes of the linear part of (II-2) and then make a linear transformation so that the coupled mode equations read:

$$\frac{d\beta_i}{dt} = \mu_i\beta_i + D_i'' + \varepsilon\Gamma''^{i}_{jk}\,\beta_j\beta_k \qquad \text{(II-3)}$$

in which the β's, the D'''s and the Γ'''s are linear combinations of the μ''s, the D''s and the Γ's. In the case of our example the equations are

$$\frac{d}{dt}\left(x + \frac{v}{i\omega}\right) = i\omega\left(x + \frac{v}{i\omega}\right) - \frac{\varepsilon}{i\omega}v|v| + \frac{D}{i\omega}\sin\Omega t$$

and its complex conjugate.

We denote the solution of:

$$\frac{d\beta_i}{dt} = \mu_i\beta_i + D_i'' \qquad \text{(II-4)}$$

by LOW (I) where:

$$\text{LOW}(I) = \beta_i(o)e^{\mu_i t} + \int_o^t e^{\mu_i(t-t')} D_i''(t')dt' \qquad \text{(II-5)}$$

once again our example would give

$$\text{LOW} = \left(X_o + \frac{V_o}{i\omega}\right)e^{i\omega t} + \frac{e^{i\omega t}}{i\omega}\int_o^t D\sin\Omega t' e^{-i\omega t'}dt'$$

From this we write the next term in the formal perturbation expansion of the solution of (II-4) in the next order as:

$$\beta^{(2)} = \int_0^t e^{\mu_i(t - t')} \sum_{jk} \Gamma^i_{jk} \text{ LOW(J)·LOW(K)} dt' \tag{II-6}$$

Inserting the lowest order results into this equation we have:

$$\beta_i^{(2)} = \int_0^t \exp[\mu_i(t - t')] \sum_{j,k} \exp[(\mu_j + \mu_k)t'] \; . \tag{II-7}$$

$$\Gamma^{i}_{jk} \left[\beta_j(o) + \int_0^{t'} e^{-\mu_j\tau} D_j''(\tau)\, d\tau\right] \cdot \left[\beta_k(0) + \int_0^{t'} e^{-\mu_k\tau} D_k''(\tau) d\tau\right]$$

Next we can substitute the expansion

$$D_i = \sum_\rho D_{i\rho} e^{i\Omega_\rho t}$$

into (II-7) and then carry out the integrations straight-forwardly. But if we were to do so a classic difficulty might be encountered; namely that the frequency of one or more of the terms on the right-hand side of (II-6) is equal to μ_i. Such a term is called secular and when integrated leads to a term that goes linearly with time and hence is unbounded.

We discuss this situation in some detail in Appendix A and show how such secular terms can be treated by appropriately shifting the eigenvalues. After this has been done one can carry out the integrations indicated in (II-7).

Returning to the example of the nonlinearly damped oscillator that was introduced above we shall apply the procedures (the BKM method) given in Appendix A. First, consider the case of the undriven oscillator. We have then, with $A = x + v/(i\omega)$

$$\frac{dA}{dt} = i\omega A - \frac{\varepsilon}{i\omega} v|v|$$

The lowest order solution is then $A = e^{i\psi}$, $\psi = \omega t$. Assuming that $A = ae^{i\psi} + \epsilon u(a, \psi) + \ldots$ and that

$$\frac{d\psi}{dt} = \omega + \epsilon\, B_1(a) + \cdots$$

$$\frac{da}{dt} = \epsilon\, C_1(a) + \ldots$$

one finds that $B_1 = 0$, $C_1 = -\frac{4}{3} a^2 \omega$. Here we have used the BKM method. The implied change in the decay rate of the lowest order solution is the device used to counter the terms proportional to $e^{i\psi}$ would arise upon inserting $A = ae^{i\psi}$ and its complex conjugate into the term $-\epsilon v|v|$.

This procedure can be carried one step further by means of the so-called method of equivalent linearization. The results that have just been obtained indicate that the amplitude damps at a rate given by

$$\frac{1}{a}\frac{da}{dt} = -\frac{4}{3}\epsilon a\omega$$

Let $\bar{a}$ denote the average amplitude corresponding to some given initial condition. Then we write the equation for a linear oscillator that is in some sense equivalent to the nonlinear one as:

$$\frac{dv}{dt} + \frac{8}{3}\epsilon\bar{a}\omega v + \omega^2 x = 0$$

The solution of this equation is, to first order in

$$x = x(0)e^{-4/3\ \epsilon\bar{a}t}\cos(\omega t + \phi)/\cos(\phi)$$

with ϕ to be determined by initial conditions.

In the case of the driven oscillator we see that the effective damping rate will depend on the driving forces since these determine the average amplitude. If we were to ignore the nonlinear term and look for the steady state of the oscillation under the driving force we would find that

$$x = \frac{D \sin\Omega t}{\omega^2 - \Omega^2}$$

Taking $\bar{a} = D/|\omega^2 - \Omega^2|$ we could write the complete equation for the oscillator as

$$\frac{dv}{dt} + \frac{8}{3}\frac{\varepsilon\bar{a}\Omega}{\pi} v + \omega^2 x = D \sin\Omega t - \varepsilon v|v| + \frac{8}{3}\frac{\varepsilon\bar{a}\Omega}{\pi} v$$

The method of equivalent linearization chooses the coefficient of v on the LHS of the above equation just so that the effects of the last two terms on the right hand side of the equation cancel out when averaged over a cycle. Then, in fact, if one were interested only in the response at the frequency Ω one would write

$$\frac{dv}{dt} + \left(\frac{8}{3}\frac{\varepsilon D\Omega}{\omega^2 - \Omega^2}\right) v + \omega^2 x = D \sin\Omega t \ .$$

The principle result from the analysis has been the determination of an effective linear damping coefficient, one that is dependent on the amplitude and the frequency of the driving forces.

B. Implementation

Almost all of the analytical processes mentioned in Section A have been implemented by programs written in PL/1 FORMAC. This includes the processes of calculation of the secular determinant of the linear part of the system, linearization, diagonalization, perturbation expansion, and substitution of the second order results back into the differential equations. In essence only the last steps involved in eigenvalue re-evaluation have not been completely implemented.

The listings of these programs is given in Appendix B. The program names and their functions are

FIRST	shifts the points of linearization.
SECOND	Finds eigenvalues and left-and-right eigenvectors.
THIRD	Carries out diagonalization formation of lowest order solution and parts of perturbation expansion.

Of these three programs FIRST and THIRD have been translated into MACSYMA. It is probably better to perform the function of SECOND by a numerical program since the eigenvalues cannot be found analytically very readily even for a system with five modes. The programs run as listed; they are not complete in the sense that our ideas are still developing.

We have also "boiler-plated" a program together which numerically integrates ordinary differential equations, plots the solutions, and also Fourier analyzes these solutions. We have used this program to investigate the qualitative nature of the solutions so that we can begin the analytical approximation of the solutions.

III. Model for Saturation of the Baroclinic Instability.

In this section a pair of coupled nonlinear equations describing the time evolution of a baroclinic instability. This model is completely solvable and we can illustrate how our perturbative approach would succeed in approximating the correct solution. We use the same model as Saltzman and Tang[5] do in their perturbative calculation of this instability including these features:

1) a two level model on a beta plane of finite width (a beta-strip) is employed, with x in the east-west direction and y measured north-south.
2) the thermal wind relation is taken to be $f\psi = \Phi$.
3) we look for a wave solution of the form

$$\left.\begin{matrix}\psi_1 \\ \psi_3 \\ \omega_2\end{matrix}\right\} \sim e^{ikx} \sin \ell y$$

which vanishes at the two edges of the beta-strip.

4) the base flow has a thermal wind given by $U_T = (U_1 - U_3)/2$ and a mean flow given by $U_M = (U_1 + U_3)/2$ both of which are in the east-west direction.

The equations governing the first order eddy fields (the stream function) can be written as

$$\frac{d}{dt}\begin{pmatrix}\psi_M \\ \psi_T\end{pmatrix} = ik\begin{pmatrix}-R_M & U_T \\ (2r-1)U_T & -R_I\end{pmatrix}\begin{pmatrix}\psi_M \\ \psi_T\end{pmatrix} \qquad \text{III-1}$$

where $\psi_M = (\psi_1 + \psi_3)/2$; $\psi_T = (\psi_1 - \psi_3)/2$. R_M and R_T are the speeds of the two Rossby waves which are the solutions of these equations when $U_T = 0$. The quantity r is given by $[1 + (k^2 + \ell^2)/\lambda^2]^{-1}$ where λ is the characteristic reciprical length $f/\sqrt{\sigma\Delta p}$.

It is convenient to introduce new amplitudes by means of the substitution $\psi = \psi' e^{ik[(R_M + R_T)/2]t}$ and then dropping the primes. One gets

$$\frac{1}{ikU_T}\frac{d}{dt}\begin{pmatrix}\psi_M\\ \psi_T\end{pmatrix} = \begin{pmatrix}\alpha & -1\\ (2r-1) & -\alpha\end{pmatrix}\begin{pmatrix}\psi_M\\ \psi_T\end{pmatrix} \qquad \text{(III-2)}$$

where

$$\alpha = \frac{df/dy}{\ell^2 + k^2}\frac{r}{2}$$

The eigenvalues of the matrix on the right hand side of equation III-2 given by

$$\lambda = \pm\, i\sqrt{(2r-1) - \alpha^2}$$

Denoting by P the quantity $-\, i\sqrt{(2r-1) - \alpha^2}$ (so that $ikP\, U_T$ is the positive growth rate of this baroclinic instability in the limit of vanishing amplitude) we introduce the linear combinations $\phi_\pm \equiv \psi_M - \psi_T/(\alpha \pm P)$ for which

$$\frac{d}{dt}\phi_\pm = \pm\, ikU_T P$$

Now the results of the S-T calculation in second-order show that there is a mean zonal flow which varies as sin(2ly) and which has equal and opposite

values on the first and third levels. If we denote the amplitude of this zonal flow, (call it the 2nd harmonic flow) by -2Ψ on level 1 and $+2\Psi$ on level 3 then we find that (ignoring α compared to P):

$$\frac{d\phi_+}{dt} = ikU_T P \left(1 + \frac{\Psi}{P^2} \phi_+ r\right) \qquad \text{(III-3)}$$

The second harmonic flow in turn is fed by the nonlinear terms in the vorticity equation; its rate of increase is determined by the product $\psi_1 \psi_3$. Within the context of the perturbation calculation then we have

$$\begin{aligned} \frac{d\phi_+}{dt} &= (\nu - \lambda\phi_+\Psi) \\ \frac{d\Psi}{dt} &= K\phi_+^2 \end{aligned} \qquad \text{(III-4)}$$

where K is a positive constant and depends on the other parameters (k, μ, f, etc.) of the system, $\nu = kU_T|P|$ and $\lambda = kU_T r/P^2$. In this last equation we should include terms proportional to ϕ_-^2 and to $\phi_+ \phi_-$; however, during the growth of the instability these terms will be relatively small in comparison with ϕ_+^2.

One finds the solution to this set of equations to be given by

$$\phi_+ = \frac{\phi_+(0) \cosh(\sqrt{C}\, T)}{\cosh(\sqrt{C}\,(T - t))} \qquad \text{(III-5)}$$

where $\sqrt{C} \simeq \nu$ and $e^{-2\sqrt{C}\,T} \simeq \lambda K\phi_o^2/8C << 1$. T is the time for saturation.

Since $e^{+\sqrt{C}\,T}$ is large we can write the solution given above as

$$\phi \simeq \frac{\phi(0)e^{\nu t}}{1 + e^{-2\nu T}e^{2\nu t}} = \phi(0)\ \left(e^{\nu t} - e^{3\nu t}\ e^{-2\nu T} + \ ..\right) \qquad \text{(III-6)}$$

for $t \ll T$. The second term $\sim e^{3\nu t}$ would be the perturbation result. Presumably one might be able to work backwards from perturbation theory applied to a growing mode. We also find that

$$\Psi = \frac{\nu}{\lambda}\ (1 + \tanh\ [\sqrt{C}\ (t - T)]) \qquad \text{(III-7)}$$

and $\frac{\dot{\phi}}{\phi}$ = growth rate $= -\ \nu\ \tanh\ [\sqrt{C}\ (t - T)]$ (III-8)

The results indicate that at saturation, i.e. when $d\phi/dt \rightarrow 0$ one has

$$\left.\begin{aligned} \phi_{max} &= \frac{2\sqrt{2}}{\lambda K}\ \nu \\ \Psi_{max} &= 2\nu/\lambda \end{aligned}\right\} \qquad \text{(III-9)}$$

It is seldom possible to find an exact solution; in most cases one must use perturbation theory. Perturbation of a growing mode yields terms that increase faster than the lowest order solution (see Equation III-6 above for an interpretation of such terms). In terms of the formalism discussed in the previous section which we intend to use for perturbation calculations one rewrites equations (III-4) in terms of the variable

$$\begin{aligned} \phi_+ &= \phi_{max/2} + \phi' \\ \Psi &= \frac{\nu}{\lambda} + \psi' \end{aligned} \qquad \text{(III-10)}$$

to obtain

$$\frac{d\phi'}{dt} = -\lambda\psi' \frac{\phi_{max}}{2} - \lambda\phi'\psi'$$

$$\frac{d\psi'}{dt} = 2K \frac{\phi_{max}}{2} \phi' + K\phi'^2 + K \frac{\phi_{max}^2}{4} \tag{III-11}$$

If we linearize these equations with respect to $\phi' = \psi' = 0$ the lowest order solution would be an oscillation at the frequency given by $\omega^2 = 2K\lambda \frac{\phi_{max}}{2}^2 = 2\nu^2$. At this point one could then apply the BKM method. In Figure 1 we sketch the relationship between the exact solution and the appropriate solution of the linearized version of (III-11).

IV. Further Examples

We illustrate the method outlined in Section II by means of a relatively simple but non-trivial problem. Consider three modes that are coupled together and which are driven by a external driver in the following fashion:

$$\frac{dA}{dt} = BC + D\sin(\Omega t);$$

$$\frac{dB}{dt} = -AC; \qquad \text{(IV-1)}$$

$$\frac{dC}{dt} = -k^2 A B; \quad k^2 < 1;$$

In the absence of the driver the solution of these equations (with $A(0) = 0$, $B(0) = 1$, $C(0) = 1$) would be:

$$A = \mathrm{sn}(t, k)$$

$$B = \mathrm{cn}(t, k)$$

$$C = \mathrm{dn}(t, k)$$

where sn, cn, and dn are the usual Jacobi elliptic functions. We consider the driven case here. As a first step we integrated this set of equations numerically using the initial values: $A = 0$, $B = 1$, $C = 1$. The numerical solutions indicated that the mode amplitude C had a non-zero average value close to 1 while the amplitudes A and B averaged to zero. Therefore, we linearized the equations about the "point" $A = 0$, $B = 0$, $C = 1$ using the program FIRST (See Section II). The matrix of the linear part of the new equations is

$$\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Next we solved the secular determinant and formed the left and right eigenvectors using program SECOND. The crank was turned further and in Figure 2 we show some of the results of substituting the perturbation solution back into the differential equation.

Even in this relatively simple example we find that very many terms could result from these simple calculations.

The BKM method permits selection of those terms that lead to changes in the growth rate and frequency of the oscillations (waves).

In order to test "experimentally" if there was any validity to the methods we have proposed in Section III we integrated the equations IV-1 numerically.

At the same time we also integrated the following linearized equations numberically.

$$\frac{dA}{dt} = 1/\sqrt{1 + k^2/4} \;\; B + D \sin\Omega t$$

$$\frac{dB}{dt} = - 1/\sqrt{1 + k^2/4} \;\; A$$

In both cases $D = .5$, $k^2 = 0.5$, and $\Omega = 0.2$.

The point of this calculation was to determine if the solution of <u>linearized</u> equations approximates the exact solution. (The frequency correction, given by the reciprocal square root, is the standard first order result for elliptic functions). Figures 3 and 4 show the results for A and B ; the correspondence is sufficient to indicate that there is merit in the approach.

We have also just begun an application of the methods discussed in previous sections to the modal equations derived by E. N. Lorenz[6] to model the rotating basin experiments. We have integrated these equations numerically for certain values of his frictional and heating parameters as well as for initial conditions. On the basis of these numerical results we carried out some of the analytic processes described in Section 3 by means of the programs that have been developed. Figure 5 shows the results some typical numerical results. This work is continuing.

ACKNOWLEDGEMENT.

I would like to thank Col. John Perry of ARPA for general guidance in meteorology, Dr. Larry Gates of the RAND Corporation for some trenchant comments about our general approach, and Dr. Ernest Valeo, formerly at Princeton Plasma Physics Laboratory, for useful discussions on non-linear coupled mode problems.

Appendix A. Slow Time Scale Equations

In this appendix we shall give a brief description of the Bogoliubov-Krylov-Mitropolsky (BKM) scheme for deriving slow time scale equations whose solutions are "asymptotic" to the solution of a nonlinear differential equation. After discussing the BKM for an oscillator with one degree of freedom in the conventional manner we introduce the concepts connected with the method of equivalent linearization. The use of complex notation is then introduced as this is more natural for the equations we consider. Finally we discuss the case of many degrees of freedom.

The nonlinear equations that we look at first are of the sort that arise in the mechanical oscillations of a system, viz

$$m \frac{d^2x}{dt^2} + kx = \varepsilon m f(x, \dot{x}); \; \dot{x} \equiv \frac{dx}{dt} ; \tag{A-1}$$

wherein we are considering a system with one degree of freedom and where f is a (non-linear) function of x and $\dot{x}$. Furthermore is a parameter which is to be considered small. The BKM scheme yields solutions in powers of ε which are asymptotic to the solution of (A-1) in the mathematical sense, i.e. a finite number of the terms in the expansion well approximates the true solution.

First consider a simple perturbation solution in the case so that (A-1) becomes

$$\frac{d^2x}{dt^2} + \omega_o^{\,2}x = -\,\varepsilon x^3, \; \omega_o^{\,2} = \frac{k}{m} \tag{A-2}$$

If we assume that we can write the solution as

$$x = \sum_{n=0} \varepsilon^n x^{(n)}(t) \tag{A-3}$$

we obtain, upon inserting Equation (3) into (2) and equating terms of equal order in ε

$$\frac{d^2x^{(0)}}{dt^2} + \omega_o^2 x^{(0)} = 0 \tag{A-4-a}$$

$$\frac{d^2x^{(1)}}{dt^2} + \omega_o^2 x^{(1)} = - \varepsilon x^{(0)^3} \tag{A-4-b}$$

$$\vdots$$

This is the so-called Poisson method and it leads to immediate difficulties. The general solution of (A-4-a) is

$$x^{(0)} = B \cos(\omega_o t + \phi)$$

where B and ϕ are constants. Now then

$$x^{(0)^3} = \frac{3 B_o^3}{4} \cos(\omega_o t + \phi) + \frac{B_o^3}{4} \cos(3\omega_o t + 3\phi)$$

This means that the right hand side of the equation for $x^{(1)}$ has a term $(3B_o^3/4 \cdot \cos(\omega_o t + \phi))$ which is a solution of homogeneous equation. As is well known the particular integral will then have a term that behaves as $t \sin(\omega t)$. This is the so-called secular behavior which is unbounded in time.

The physics of the problem illuminates the mathematical solution. The non-linear restoring forces much change the frequency of the oscillation

because the potential energy is not simple $kx^2/2$ here but rather is

$$PE = kx^2/2 + \varepsilon m x^4/4$$

Consequently the solution is approximately given by

$$B\cos(\omega_o t + \delta\omega t + \phi) = B\cos(\omega_o t + \phi) - \delta\omega\cdot t\cdot B\sin(\omega_o t + \phi)$$

with $\delta\omega$ arising from the non-linearity. While the names of Lindstedt and Poincare' are associated with methods for obtaining period solutions of A-1 we shall describe here the method due to Bogoliubov, Krylov and Mitropolsky; the method is somewhat beuristic and intuitive but powerful and direct.

One starts from the fact that the solution to A-1 with $\varepsilon = 0$ is $a\cos\psi$ with $\frac{da}{dt} = 0$, (a = constant) and $\frac{d\psi}{dt} = \omega_o$, $\psi = \omega_o t + \phi$.

We recognize that the nonlinear terms may cause a change in the amplitude, a , on a slow time scale and in the frequency, $d\psi/dt$. We write then

$$x = a\cos\psi + \varepsilon\, U^{(1)}(a, \psi) + \ldots \qquad \text{(A-5-a)}$$

$$\dot{x} = -\omega_o a\sin\psi + \ldots \qquad \text{(A-5-b)}$$

$$\frac{d\psi}{dt} = \omega_o + \varepsilon\, C^{(1)}(t) + \ldots \qquad \text{(A-5-c)}$$

$$\frac{da}{dt} = \varepsilon\, A^{(1)}(t) + \ldots \qquad \text{(A-5-d)}$$

where $C^{(1)}$ and $A^{(1)}$ are to be determined so that $U^{(1)}$ will contain no secular terms. One finds then

$$\frac{d\psi}{dt} - \omega_o = - \frac{\varepsilon}{2\pi\omega a} \int_o^{2\pi} f\ (a \cos \psi, - a\ \omega_o, \sin \psi) \cos \psi\ d\psi \qquad \text{(A-6-a)}$$

$$\frac{da}{dt} = - \frac{\varepsilon}{2\pi\omega} \int_o^{2\pi} f\ (\cos \psi, - a\ \omega_o \sin \psi) \sin \psi\ d\psi \qquad \text{(A-6-b)}$$

Returning to the example we started from in which

$$f(x, \dot{x}) = mx^3 = ma^3 \cos^3 \psi$$

we find

$$\frac{d\psi}{dt} = \omega_o - \frac{\varepsilon m a^2}{2\pi\omega} \int_o^{2\pi} \cos^4 \psi\ d\psi = \omega_o - \frac{\varepsilon m a^3}{2\pi\omega}$$

$$\frac{da}{dt} = 0$$

We can physically interpret the results given by A-6 by recognizing that the integral in (A-6-b) is the work done per cycle while that in (A-6-b) involves the so-called reactive power.

B. The Method of Equivalent Linearization

There is another way of interpreting the results given above using a somewhat cruder approximation. Recall that we started by looking at

$$\frac{d^2x}{dt^2} + \omega_o{}^2 x = - m\varepsilon f(x, \dot{x})$$

The lowest order BKM solution (the $\varepsilon U^{(1)}$ term can be ignored here) is:

$$x = a \cos \psi$$

with

$$\frac{da}{dt} = \varepsilon A^{(1)}(a)$$

$$\frac{d\psi}{dt} = \omega_o + \varepsilon C^{(1)}(a)$$

We approximate these last two equations as

$$\frac{1}{a}\frac{da}{dt} = \left(\frac{\varepsilon A^{(1)}(a)}{a}\right)_{a = a_o}$$

$$\frac{d}{dt} = \omega_o \left(1 + \frac{\varepsilon C^{(1)}}{\omega_o}\right)\Bigg|_{a = a_o}$$

where a_o is the amplitude for $t = o$. One observes that the corresponding solution for x could have been obtained from the solution of the differential equation

$$\frac{d^2x}{dt^2} + 2\bar{\lambda}x + \frac{\bar{k}}{m}x$$

if to the first order in ε we had chosen

$$\lambda = -\frac{\varepsilon A^{(1)}(a)}{a}\bigg|_{a_o}$$

$$\frac{\bar{k}}{m} = \omega_o^{\;2} + 2\omega_o\,\varepsilon C'(a)$$

We note this use of perturbation theory to modify the <u>eigenvalues obtained</u> by linearization.

C. Complex Formulation.

The previous results can also be viewed from the results on first order matrix equations. The second order differential Equation A-2 can be written in matrix form as (use $P = \dot{x}/m$)

$$\frac{d}{dt}\begin{pmatrix} X \\ P \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{m} \\ -k & 0 \end{pmatrix}\begin{pmatrix} X \\ P \end{pmatrix} + \begin{pmatrix} 0 \\ \varepsilon f(x, \dot{x}) \end{pmatrix}$$

Introducing $\alpha = x + P/im\omega_o$ and its complex conjugate α^* as new variables we obtain

$$\frac{d\alpha}{dt} = i\omega_o\alpha + \frac{\varepsilon}{im\omega_o} f(\alpha, \alpha^*)$$

and its complex conjugate. Here $\omega_o = \sqrt{k/m}$

Writing $\alpha = ae^{i\psi} + \sum_{1}^{\infty} \varepsilon^n \beta^{(n)} (a, \psi)$

along with

$$\frac{d}{dt} = \omega_o + \sum_{n=1}^{\infty} \varepsilon^n \psi^{(n)} (a)$$

$$\frac{da}{dt} = \sum \varepsilon^n A^{(n)} (a)$$

one obtains

$$-\omega_o \frac{\partial}{\partial\psi} (\beta^{(n)} e^{-i\psi}) =$$

$$-(A^{(n)} + i\psi^{(n)}a) + e^{-i\psi}\left[F^n - \sum_{s=1}^{n-1} (A^{n-s}\frac{\partial}{\partial a} + \psi^{n-s}\frac{\partial}{\partial\psi})\beta^{(s)}\right]$$

F^n is the coefficient of ε^n in the development of $m\varepsilon f(x, \dot{x})/i\omega$.

In order to avoid secularities one must have

$$A^{(n)} + i\Psi^{(n)}a \equiv \frac{1}{2\pi} \int_0^{2\pi} e^{-i\psi} (F^n - \sum_{s=1}^{n-1} (A^{n-s} \frac{\partial}{\partial a} + \psi^{n-s} \frac{\partial}{\partial \psi}) \beta^{(s)}$$

D. Several Degrees of Freedom

In abstract form we must solve

$$\frac{dB_i}{dt} = \sum M_{ij} B_j + F_i (\vec{B}), \; i = 1, \ldots N$$

Take S to be the matrix that diagonalizes M, i.e. $S^{-1}MS = \Lambda$ where Λ is diagonal. Take $U_e = S^{-1}_{ej} B_j$. Then

$$\frac{dU_k}{dt} = i \lambda_k U_k + \sum_i S^{-1}_{ki} F_i$$

Again, assuming $U_k = a e^{i\psi} + \varepsilon \beta^{(1)} (a, \psi)$,

$$\frac{d\psi}{dt} = \lambda_k + \varepsilon \Psi_1 (a) ,$$

and

$$\frac{da}{dt} = \varepsilon A_1 (a)$$

one obtains

$$A_1 + i \Psi_1 a = \frac{1}{2\pi} \int_0^{2\pi} \sum_i e^{-i\psi} S^{-1}_{ki} F_i \, d\psi$$

for the eigenvalue corrections to the k-th mode.

Appendix B. Program Listings

```
 INPUT TO FORMAC PREPROCESSOR
FIRST:PROCEDURE OPTIONS(MAIN);   FORMAC_OPTIONS;

/* BASED PARTLY UPON DECK FOR PERTURB ROUTINE            */
/* STARTED JUNE 27 1972              */
/* CORRECT PERTURB DECK */
/*1.*/    CALL INPUT;
/*2.*/    CALL LINEARIZE;
/*3.*/    CALL FORM_DETERMINANT;
/*4.*/    CALL OUTPUT_ALL;


/*........................ SUBROUTINES ...........................*/



OPTSET(PRINT);
INPUT:PROCEDURE;
/*1.*/ GET FILE(CONTROL)DATA (MODES,
       NOTHING);
LET(FNC(SINE)=-#I/2*EXPON.(#I*$(1))+#I/2*EXPON.(-#I*$(1));
 -   FNC(COSINE)=1/2*EXPON.(#I*$(1))+1/2*EXPON.(-#I*$(1)) );
OPTSET(EXPND);

/*2.*/ /* ZERO THE ARRAYS OF COEFFICIENTS */
       /*A*/ CALL FORMAC_ERASE('D(I)',MODES,1);
       /*B*/ CALL FORMAC_ERASE('M(I,J)',MODES,2);
       /*C*/ CALL FORMAC_ERASE('GAMMA(I,J,K)',MODES,3);
       /*D*/ CALL FORMAC_ERASE('BO(I)',MODES,1);

FORMAC_ERASE:PROCEDURE(A,N,RANK);
DCL A CHAR(*),(N,RANK,NT(4) ) FIXED BINARY;
 DO I=1 TO 4; IF I<=RANK THEN NT(I)=N; ELSE NT(I)=1; END;
ZERO: DO I=1 TO NT(1);CET(I); DO J=1 TO NT(2);CET(J);
DO K=1 TO NT(3); CET(K); DO L=1 TO NT(4); CET(L);
RET(A=0); END      ZERO; ATOMIZE(I;J;K;L); END FORMAC_ERASE;

/*3.*/  /* INPUT THE ARRAYS*/

       /*A*/ CALL EQUATIONS('DRIVER');
       /*B*/ CALL EQUATIONS('MARRAY');
       /*C*/ CALL EQUATIONS('GAMMAS');

EQUATIONS:PROCEDURE(A); DCL A CHAR(*);
ON ENDFILE(SYSIN) BEGIN; CLOSE FILE(SYSIN); GOTO EOF; END;
OPEN FILE(SYSIN) TITLE(A) INPUT;
LOOP: GET FILE(SYSIN) LIST(VALUE)COPY; FORM(VALUE); GOTO LOOP;
EOF: RETURN; END EQUATIONS;


DCL FORMAC_ERASE ENTRY(CHAR(15)VAR,FIXED BIN,FIXED BIN),
   EQUATIONS ENTRY (CHAR(6)VAR);
EOSR: END INPUT;


LINEARIZE : PROCEDURE;
 IMODES: DO I=1 TO MODES; CET(I);
LET(SUM(1)=0; SUM(2)=0);
```

```
    JMODES: DO J=1 TO MODES; CET(J);
         LET(SUM(1)=0);
         DO K=1 TO MODES; CET(K);
         LET(SUM(1)=SUM(1)+GAMMA(I,J,K)*BO(K)+GAMMA(I,K,J)*BO(K);
              SUM(2)=SUM(2)+GAMMA(I,J,K)*BO(J)*BO(K) );
         END;
   LET(SUM(2)=SUM(2)+M(I,J)*BO(J));
    LET(M(I,J)=M(I,J)+SUM(1));       ATOMIZE(SUM(1));
    END JMODES;
LET(D(I)=D(I)+SUM(2)); ATOMIZE(SUM(2));
END IMODES;
EOSR: RETURN; END LINEARIZE;



OUTPUT_ALL:PROCEDURE;
/*A*/ CALL DISK('D(I)',MODES,1,'DRIVE');
/*B*/ CALL DISK('GAMMA(I,J,K)'  ,MODES,3,'GAM');
NOTEST='1'B;
/*C*/ CALL DISK('ROW(I)',MODES,1,'ROWS');
RETURN;
END OUTPUT_ALL;


DCL DISK ENTRY(CHAR(20)VAR,FIXED BINARY,FIXED BINARY,CHAR(20)VAR);
DISK:PROCEDURE(A,N,M,B);    OPEN FILE(PUNCH)OUTPUT TITLE(B);
DCL (A,B) CHAR(*),(N,M,NT(4)) FIXED BINARY;
        DO I=1 TO 4; IF I<=M THEN NT(I)=N; ELSE NT(I)=1; END;
CARDS:
 DO I=1 TO NT(1); CET(I); DO J=1 TO NT(2); CET(J);
DO K=1 TO NT(3);CET(K); DO L=1 TO NT(4); CET(L);
  IF NOTEST THEN GOTO PUNCHER;
IF ¬DENFMCG('Z9999997='||A,'Z9999998=0') THEN  PUNCHER:DO;
 CALL DENFMCH(FORMULA,A);
PUT FILE(PUNCH)LIST(FORMULA)SKIP;
END;
END CARDS;
CLOSE FILE(PUNCH);
END DISK;
FORM_DETERMINANT:PROCEDURE;
DO I=1 TO MODES;CET(I); LET(ROW(I)=M(I,I)-MU);
DO J=1 TO MODES; CET(J);
IF J<I THEN
LET(ROW(I)=(M(I,J),ROW(I)));
IF J>I THEN LET(ROW(I)=(ROW(I),M(I,J)));
END;
END;
END FORM_DETERMINANT;


DCL
        FORMULA CHAR(800)VAR,
        ITERATE# FIXED BINARY,
        NOTEST BIT(1) INITIAL('0'B),
        MODES FIXED BINARY,
        DENFMC3 ENTRY(FIXED BIN(31,0),FIXED BIN(31,0))EXTERNAL,
        VALUE CHAR(100) VAR,
        RANK FIXED BINARY,
        NOTHING FIXED;
```

```
END_OF_PROGRAM:
END FIRST;
```

```
(CHECK(EIGENVECTORS,SOLVE,SEGREGATE,LEFT_EIGENVECTORS)):
SECOND: PROCEDURE OPTIONS(MAIN); FORMAC_OPTIONS;


/* JUNE 27,1972 DIAGONALIZATION OF LINEARIZED COUPLED MODE EQUATIONS
    ADAPTED FROM PORTIONS OF THE CARLMAN PROGRAM              */





/*1..................*/      CALL INPUT;
/*2..................*/    CALL DETERMINANT(DIMENSION);
/*2A################*/     CALL SEGREGATE;
/*2B.NNNNNNNNNNNNNNN*/      CALL NUMERICAL_EVALUATION;
/*3.>>>>>>>>>>>>>>>>*/ CALL   RIGHT_EIGENVECTORS;
/*4.<<<<<<<<<<<<<<<<*/ CALL LEFT_EIGENVECTORS;
/*5..................*/  CALL EIGENVALUES;



INPUT:PROCEDURE; ON ENDFILE(SYSIN) GO TO EOF;
GET DATA(NROW); DIMENSION,@DIM=NROW;
DCL PAIR CHAR(200) VAR;
      GET FILE(PAIRS)LIST(PAIR)COPY; CET(PAIR);  CLOSE FILE(PAIRS);
LET(@TIMES=0);
CET(@DIM;NROW); LET(Q$(0)=0;TYPE=0);
LET(PROD(1)=1);
 LET(Q$(-1)=0; TYPE=-1);
NDIM,@DIM,NROW=DIMENSION;
CET(@DIM ;NDIM);
LET(I(@DIM)=1);
DO J=1 TO @DIM; NUMBER(1,J)=J; END;
DO I=0 TO NROW;CET(I);LET(N(I)=0);END;
 CLOSE FILE(SYSIN); OPEN FILE(SYSIN)TITLE('ROWS');
DO I=1 TO NROW;CET(I); GET LIST(RESULTS)COPY; FORM(RESULTS);
DO J=1 TO NROW; CET(J); LET(TERM(I,J)=ARG(J,ROW(I)));
LET(TERM(I,J)=EVAL(TERM(I,J),PAIR));
END; END;
PUT FILE(SYSCP)EDIT('  SECULAR DETERMINANT')(SKIP,A);
GO TO EOF;
OUTPUT:ENTRY;  CHAREX(RESULTS=X(TYPE,Q$(TYPE)));
GO TO PUNCHER;
OUTPUT#:ENTRY(NAME); DCL NAME CHAR(*); CALL DENFMCH(RESULTS,NAME);
PUNCHER:
PUT FILE(SYSCP)LIST(RESULTS)SKIP;
RETURN;
PUT FILE(SYSCP) EDIT(RESULTS)(SKIP,X(1),A);
EOF: RETURN; END INPUT;


DETERMINANT:PROCEDURE(NDIM) RECURSIVE;
DECLARE (NDIM,I) FIXED BINARY;


LET(@TIMES=@TIMES+1)          ;
DCL @TIMES FIXED BINARY; @TIMES =INTEGER(@TIMES);


ONE_D:
IF NDIM=1 THEN DO;
LET(NUMBER="NUMBER(@DIM,1)");
```

```
IF IDENT(TERM(@DIM,NUMBER);0) &IDENT(Q$(TYPE);0) THEN GOTO NULL;
LET( Q$(TYPE)=Q$(TYPE)+1);
LET(X(TYPE,Q$(TYPE))=PROD(@TIMES)*TERM(@DIM,NUMBER)* "SIGN");
IF ¬IDENT(X(TYPE,Q$(TYPE));0) THEN
CALL OUTPUT;
NULL: LET(@TIMES=@TIMES-1);
RETURN;
END;




CALCULATE: DO I=1 TO ( INTEGER(@DIM)-INTEGER(@TIMES)+1); LET(I="I");
LET( I(@TIMES)=I);
      LET(N="NUMBER(@TIMES,I)");
LET(PROD(@TIMES+1)=PROD(@TIMES)*TERM(@TIMES,N));
IF IDENT(PROD(@TIMES+1);0) THEN GOTO END_OF_CALCULATION;

L=0; DO J=1 TO NDIM; IF J¬=I THEN DO;
       L=L+1;NUMBER(@TIMES+1,L)=NUMBER(@TIMES,J);END;END;

CALL DETERMINANT(NDIM-1);
LET(I=I(@TIMES));
END_OF_CALCULATION:
END CALCULATE;


LET(@TIMES=@TIMES-1);
RETURN; END DETERMINANT;






SIGN:PROC    RETURNS(CHAR(4)); LET(SIGN=+1);
DCL N$    (7) FIXED BINARY;
DO L#=1 TO NROW;CET(L#); N#=INTEGER(I(L#)); N$(L#)=NUMBER(L#,N#);END;
/*
PUT EDIT((N$(K$) DO K$=1 TO NROW) ) ( 10 F(3,0) );
*/

DO L#=1 TO NROW; IF N$(L#)¬=L# THEN DO J#=L#+1 TO NROW;
   IF  N$(J#)=L# THEN DO;
           N$(J#)=N$(L#);LET(SIGN=-SIGN); END;
       END;
 END;
RETURN('SIGN'); END SIGN;

EIGENVECTORS:PROCEDURE;
RIGHT_EIGENVECTORS:ENTRY;
PUT FILE(SYSCP)EDIT(' RIGHT EIGENVECTORS')(SKIP,A);
     EIGENVECTORS#:ENTRY;
CET(LEFT);
LET(X00=1);
DO J=2 TO DIMENSION; CET(J);
LET(D(J-1)=-X00*TERM(J,1));     /* USUALLY X00-->1; MAY -->0 */
END;
MAKE_SMALLER: DO I=2 TO DIM;CET(I); DO J=2 TO DIM; CET(J);
LET(TERM(I-1,J-1)=TERM(I,J)); END; END MAKE_SMALLER;
DIMENSION=DIMENSION-1;
@DIM,NROW=DIMENSION;
CET(@DIM);      LET(I(@DIM)=1);
IF ON THEN DO;
ON =¬ON;
         PUT FILE(SYSCP)EDIT(' DETERMINANT FOR EIGENVECTORS')(SKIP,A);
LET(TYPE=0); CALL DETERMINANT(NROW); END;
CALL SOLVE;
```

```
NROW,@DIM,DIMENSION=NROW+1;
END EIGENVECTORS;


LEFT_EIGENVECTORS:PROCEDURE;
LEFT=100;
PUT FILE(SYSCP)EDIT(' LEFT EIGENVECTORS')(SKIP,A);
GET(@DIM);
/* FORM THE TRANSPOSED MATRIX */
DO I=1 TO NROW;GET(I);DO J=1 TO NROW;GET(J);
LET(TERM(I,J)=EVAL(ARG(I,ROW(J)),PAIR)); END; END;
CALL EIGENVECTORS#;
END LEFT_EIGENVECTORS;


SOLVE:PROCEDURE(DIMENSION);  /* INTERFACE FOR CALLING THE DETERMINANT
                                  FOR SOLVING SIMUTLANEOUS EQUATIONS*/
DCL(I,J) FIXED BINARY;
OPTSET(PRINT);
OPTSET(NOPRINT);
VARIABLE_SOLUTION: DO J=1 TO DIM; GET(J);
REPLACE_A_COLUMN:DO I=1 TO DIM;
GET(I);
LET(TSAVE(I)=TERM(I,J));
LET(TERM(I,J)=D(I));   END REPLACE_A_COLUMN;
LET(TYPE=J+LEFT); LET(Q$(TYPE)=C); CALL DETERMINANT(@DIM);
RESTORE_PREVIOUS_COLUMN:
DO I=1 TO DIM;GET(I); LET(TERM(I,J)=TSAVE(I)); END;
END VARIABLE_SOLUTION;
EOSR:END SOLVE;


OPTSET(PRINT);


NUMERICAL_EVALUATION:PROCEDURE; DCL PAIR CHAR(200)VAR;
NRAW=NROW+1;
EOF: DO I=0 TO NROW; GET(I); P$(I+1)=ARITH(P$(I)); END;
PUT LIST((P$(I) DO I=1 TO NRAW));
EOSR: RETURN; END NUMERICAL_EVALUATION;


EIGENVALUES:PROCEDURE;
P$=P$/P$(NRAW);
DO I=1 TO NROW; Q#(I)=COMPLEX(P$(NROW-I+1),0.); END;
DO I=1 TO NROW; PUT DATA(Q#(I)); END;
CALL PRTC(Q#,NROW);
IF ERROR= OK THEN PUT LIST((Q#(I) DO I=1 TO NROW))SKIP;
IF ERROR=OK THEN PUT FILE(SYSCP) DATA((Q#(I) DO I=1 TO NROW))SKIP;
 /* FOR EACH EIGENVALUE*/


DO I=1 TO NROW;GET(I);
FLOATA(EIGEN(I)=REAL(Q#(I))); FLOATA(B=IMAG(Q#(I)));
LET(EIGEN(I)=EIGEN(I)+#I*B);
LET(MU(I)=EIGEN(I)); CHAREX(STRING=MU(I));
CALL OUTPUT#;       ATOMIZE(MU(I));
LET(DET(I)=0); DO J=1 TO INTEGER(Q$( 0)); GET(J);
LET(DET(I)=EVAL(X( 0,J),MU,EIGEN(I))+DET(I));
END;
```

```
IF IDENT(DET(I);0) THEN DO;
PUT LIST(' EIGENVECTOR ',I, 'NOT FOUND');
GOTO EIGENMODES; END;

LEFT_RIGHT: DO L=0 TO 1; CET(L); /* L=0-->RIGHT, L=1-->LEFT*/

COMPONENTS:DO J=1 TO NROW-1; CET(J); LET(JL=J+L*100);
    LET(COMPON(I,J,L)=0); NTOTAL=INTEGER(Q$(JL) );

       DO K=1 TO NTOTAL; CET(K);
          LET(COMPON(I,J,L)=COMPON(I,J,L)+EVAL(X(JL,K),MU,EIGEN(I)));
       END;

LET(COMPON(I,J,L)=COMPON(I,J,L)/DET(I));
LET(COMP(I,J,L)=EVAL(COMPON(I,J,L),#I,-#I));
 END COMPONENTS;


AGAIN: DO J= NROW TO 2 BY -1; CET(J); LET(
COMPON(I,J,L)=COMPON(I,J-1,L); COMP(I,J,L)=COMP(I,J-1,L)); END;
LET(COMPON(I,1,L)=1; COMP(I,1,L)=1);

LET(SQ=0);
DO J=1 TO NROW-1; CET(J);LET(SQ=SQ+COMPON(I,J,L)*COMP(I,J,L)); END;
DO J=1 TO NROW-1;CET(J);LET(COMPON(I,J,L)=COMPON(I,J,L)/SQRT(SQ)); END;

END LEFT_RIGHT;



  PUT FILE (SYSCP) SKIP;
DO J=1 TO NROW-1; CET(J);
LET(S(I,J)=COMPON(I,J,0);S1(I,J)=COMPON(I,J,1));
CHAREX(STRING=S1(I,J)); PUT FILE(SYSCP)LIST(STRING);
CHAREX(STRING=S (I,J)); PUT FILE(SYSCP)LIST(STRING);
END;
EIGENMODES:
END;

END EIGENVALUES;




SEGREGATE:PROCEDURE;
DO I=0 TO NROW+1;CET(I); LET(P$(I)=0); END;
NTERMS=INTEGER(Q$(-1)); DO N$=1 TO NTERMS; CET(N$);
IF LOP(X(-1,N$))=24 THEN DO; NSTART=1; NSTOP=NARGS(X(-1,N$)); END;
                  ELSE NSTART,NSTOP=0;
   DO N#=NSTART TO NSTOP; CET(N#); LET(BUMP=ARG(N#,X(-1,N$)));
     IF IDENT(BUMP;0) THEN GOTO NO_CONTRIBUTION;
  LET(HP=HIGHPOW( BUMP,MU); P$(HP)=P$(HP)+EVAL(BUMP,MU,1));
NO_CONTRIBUTION:END;
END SEGREGATE;


DCL      DENFMC3 ENTRY(FIXED BIN(31,0),FIXED BIN(31,0)) EXTERNAL,
         (INPUT,OUTPUT) ENTRY,
         DIMENSION FIXED BINARY,
         (ALPHA,BETA) FIXED BINARY,
SIGN ENTRY RETURNS(CHAR(4)),
RESULTS CHAR(800) VAR,
NUMBER(9,9)FIXED BINARY,
@DIM FIXED BINARY,
NROW FIXED BINARY,
         DIM FIXED BIN DEFINED DIMENSION,
```

```
        VALUE CHAR(100) VAR,
        LEFT FIXED BINARY INIT(0),
        NOTHING FIXED,
        P$(20),
   DETERMINANT ENTRY(FIXED BINARY);
   DCL ON BIT(1) INIT('1'B);
   DCL STRING CHAR(1000)VAR;
   DCL ERROR CHAR(1) EXTERNAL,Q#(20) COMPLEX BINARY FLOAT,
       OK   CHAR(1) INIT ('0');
   END SECOND;
```

```
1.=T,K65=GER,TIDE=N SPART090.7160Q  RB 28B BOJ
// EXEC EXPORT,PROGRAM=EDITY
//STEPLIB DD DSN=U.FPL.LIBRARY,DISP=SHR
//OUT DD DSN=U.ROSEN.ARPA(THREE),VOL=SER=RES103,
//     UNIT=SYSDA,
//     DISP=(OLD,KEEP)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
.NEW  THIRD
 (CHECK(FORM_NEW_DRIVER,CHANGE_NON_LINEAR,
 LOWEST_ORDER_SOLUTION,EVALUATE_NON_LINEAR_TERMS,
   FIRST_ORDER_ITERATION,SUBSTITUTE,INTEGRATE,
 EIGENVALUE_CORRECTION)):
 THIRD:PROCEDURE OPTIONS(MAIN); FORMAC_OPTIONS;

 /* NEW VERSION  9-22-72 INCLUDES FIRST ORDER ITERATION */


     /*1*/  CALL INPUT  ;
     /*2*/  CALL  FORM_NEW_DRIVER;          CALL CHANGE_NON_LINEAR;
      /*2.5*/CALL  LOWEST_ORDER_SOLUTION;
 CALL SAVER;
        /*3 */ CALL FIRST_ORDER_ITERATION;
     /*4*/  CALL  EVALUATE_NON_LINEAR_TERMS;



 FORM_NEW_DRIVER:PROCEDURE;
 DO I=1 TO MODES;CET(I); LET(SUM=C);
      DO J=1 TO MODES;CET(J);LET(SUM=SUM+S(I,J)*D(J));END;
      LET(NEWD(I)=SUM);ATOMIZE(SUM);
 END;
 DO I=1 TO MODES;CET(I);ATOMIZE(D(I));LET(D(I)=NEWD(I));
                 ATOMIZE(NEWD(I));END;
 END FORM_NEW_DRIVER;


 LOWEST_ORDER_SOLUTION:PROCEDURE;
 MODE:DO I=1 TO MODES; CET(I);
     LP=LOP(D(I));IF LP=24 THEN DO ;NSTART=1;NSTOP=NARGS(D(I));END;
                                   ELSE NSTART,NSTOP=0;LET(SUM=0);
 LET(K$=0);
   TERMS:DO N#=NSTART TO NSTOP;CET(N#);
          LET (TERM=ARG(N#,D(I)));
 IF IDENT(TERM;0) THEN      GO TO NODRIVE;
          LET(EXPOX=DERIV(LOG(TERM),TIME));
          LET(TERM=TERM/(EXPOX+#I*MU.(I));HOMOGEN=EVAL(TERM,TIME,0); );
 /*
     LET(K$=K$+1 ; LOW(I,K$)=HOMOGEN*EXP(#I*MU.(I)*TIME))       ;
 */
     LET(K$=K$+1 ; LOW(I,K$)=HOMOGEN);
 CALL TRANSMOGRIFY('LOW(I,K$)');
 CHAREX(STRING=        LOW(I,K$)); ATOMIZE(LOW(I,K$));
 PUT FILE(HOMOGEN)LIST('  '||STRING)SKIP;
     LET(B(0,I,K$)=LOW(I,K$);EXPONENT(0,I,K$)=#I*MU.(I));
   SAVE(B(0,I,K$);EXPONENT(0,I,K$));
          ATOMIZE(TERM;HOMOGEN);
 NODRIVE:
   END TERMS;
                  LET(K$(I)=K$);
      TOTAL(0,I)= INTEGER(K$);
 END MODE;
```

```
/*  CHANGE 9-26-72
1 CUT DOWN NUMBER OF TERMS IN THE FINAL ANSWER
2  PERMIT LOW SOLUTION TO APPEAR EVEN IF DRIVER IS ZERO IN THAT MODE*/
DO I=1 TO MODES;CET(I); LET(K$=1; B(0,I,1)=LOW(I);LOW(I,1)=LOW(I);
K$(I)=1;); TOTAL(0,I)=1;     END;




END LOWEST_ORDER_SOLUTION;


EVALUATE_NON_LINEAR_TERMS:PROCEDURE;
OPTSET(NOEDIT);
OPEN FILE(PUNCH)TITLE('NLTERMS')OUTPUT;
OPTSET(NOPRINT);

MODE: DO I=1 TO MODES; CET(I);
LET(MANY=0);
    DO J=1 TO  MODES; CET(J);
    DO K=1 TO MODES; CET(K);
IF J=K THEN GOTO NOSWEAT;
IF NOD=0 & ( IDENT(J;SPECIAL) |IDENT(K;SPECIAL)) THEN GO TO NOSWEAT;
NEST=J+K;
NASTY=NEST/2      ;        IF 2*NASTY=NEST THEN GOTO NOSWEAT;

IF ¬IDENT(GAMMA(I,J,K);0) THEN DO;
   LET(GAM=GAMMA(I,J,K)); ATOMIZE(GAMMA(I,J,K));
   IF LOP(GAM)=24 THEN DO; NSTART=1;NSTOP=NARGS(GAM); END;
ELSE NSTART,NSTOP=0;
DO NS= NSTART TO NSTOP; CET(NS);
LET(GAMMER=ARG(NS,GAM));
        DO L$=1 TO INTEGER(K$(J)); CET(L$);
        DO N$=1 TO INTEGER(K$(K)); CET(N$);
        LET(MANY=MANY+1;
     NONLINE(I,MANY)=GAMMER*LOW(J,L$)*LOW(K,N$));
PRINT_OUT(NONLINE(I,MANY));
CHAREX(STRING=NONLINE(I,MANY)); CALL OUTPUT;
SAVE(NONLINE(I,MANY));
        END; END;
   ATOMIZE(GAMMER);

END; ATOMIZE(GAM);

      END;
NOSWEAT:
END ; END;

LET( ##(I)=MANY);
END MODE;

ITER=2; CET(ITER); LET(KAPPA=ITER;KAPPA1=KAPPA-1); KAPPA1=ITER-1;
PRINT_OUT(KAPPA;KAPPA1); PUT DATA((TOTAL(2,IS) DO IS=1 TO MODES));
CLOSE FILE(PUNCH); OPEN FILE(PUNCH)TITLE('THREES')OUTPUT;
IMODES:DO I=1 TO MODES; CET(I);
DO IB=1 TO MODES; CET(IB); LET(I$(IB)=0); END;
JMODES: DO J=1 TO MODES; CET(J);
KMODES: DO K=1 TO MODES; CET(K);
        IF IDENT(GAMMA(I,J,K);0) THEN GOTO K_END;
        LOWER_ORDERS: DO SIGMA=0 TO KAPPA1; KSIG=KAPPA1-SIGMA;
            CET(SIGMA;KSIG);
            JTERMS: DO JN=1 TO TOTAL(SIGMA ,J); CET(JN);
             KTERMS: DO KN=1 TO TOTAL(KSIG  ,K); CET(KN);
OPTSET(EXPND);
LET(EXPONA= EXPONENT(SIGMA,J,JN)+EXPONENT(KSIG,K,KN));
```

```
LET(EXPON=REPLACE(EXPONA,MULIST ,#I,1));
OPTSET(  PRINT);
DO IE=1 TO MODES; CET(IB); IF IDENT(EXPON;MU(IB)) THEN GOTO NEW_DEAL;
END;    GOTO NO_ACCOUNT; NEW_DEAL:
      LET(I$(IB)=I$(IB)+1;
NEWTERM(I,IB,I$(IB))=GAMMA(I,J,K)*B(SIGMA,J,JN)*B(KSIG,K,KN)
                         * EXP(EXPONA));

OPTSET(NOPRINT);
ATOMIZE(EXPONA;EXPON);
SAVE(              EXPONENT(SIGMA,J,JN);EXPONENT(KSIG,K,KN);
NEWTERM(I,IB,I$(IB));B(SIGMA,J,JN);B(KSIG,K,KN)    );
CHAREX(STRING=NEWTERM(I,IB,I$(IB))); CALL OUTPUT;

NO_ACCOUNT:
           END KTERMS;
           END JTERMS;
        END LOWER_ORDERS;
K_END:
END KMODES;
END JMODES;
NEW_TOTAL(I)=INTEGER(I$);
END IMODES;
END EVALUATE_NON_LINEAR_TERMS;
OPTSET(PRINT);


EIGENVALUE_CORRECTION:PROCEDURE;
OPTSET(ERSNP);
DO I=1 TO MODES; CET(I);
DO J=1 TO INTEGER(## (I));
CET(J);
LET(NONLINE(I,J)=EVAL(NONLINE(I,J),MU.($),MU(I)/#I));
SAVE(NONLINE(I,J)); END;
OPTSET(PRINT);
DO J=1 TO NEW_TOTAL(I); CET(J);
SAVE(NEWTERM(I,J));
END;
END;
OPTSET(NOPRINT);
END EIGENVALUE_CORRECTION;

TRANSMOGRIFY:PROCEDURE(A); DCL A CHAR(*);
OPTSET( NOEXPND);
CET($$$=A); LET(TOP=NUM($$$);BOTTOM=DENOM($$$));
ATOMIZE($$$);
IF LOP(BOTTOM)=26 THEN DO;MSTART=1;MSTOP=NARGS(BOTTOM); END;
ELSE MSTART,MSTOP=0; LET(NEWBOT=1);
DO M$=MSTART TO MSTOP;CET(M$);
LET(FACTOR=ARG(M$,BOTTOM));
LET  (FAX=EVAL(FACTOR,#I,-#I));
REALS:LET(NEWBOT=EXPAND(FACTOR*FAX)*NEWBOT;TOP=TOP*FAX);
END;
OPTSET(PRINT);
LET("A"=TOP/NEWBOT);
ATOMIZE(NEWBOT;BOTTOM;TOP; FAX;FACTOR);
EOSR: END TRANSMOGRIFY;



OPTSET(NOPRINT);
CHANGE_NON_LINEAR:PROCEDURE;

MODE:
    DO I=1 TO MODES; CET(I);
    DO J=1 TO MODES; CET(J);
```

```
    DO K=1 TO MODES; CET(K);
     LET(SUM=0);

         DO MU =       1 TO MODES; CET(MU);
         DO LAMBDA = 1 TO MODES; CET(LAMBDA);
         DO NU=1 TO MODES; CET(NU);
         LET(SUM=SUM+GAMMA(MU,LAMBDA,NU)*S1(LAMBDA,J)*S1 (NU,K)*
                        S(I,MU));

         END; END; END; ATOMIZE(MU);
        LET(NEWGAM(I,J,K)=SUM); ATOMIZE(SUM);
 END MODE;



 FANDANGO:
 DO I=1 TO MODES;CET(I);DO J=1 TO MODES;CET(J);DO K=1 TO MODES;CET(K);

 LET(GAMMA(I,J,K)=NEWGAM(I,J,K)); ATOMIZE(NEWGAM(I,J,K));
 END FANDANGO;
 CALL ATOMIC('GAMMA(I,J,K)',3);
 END CHANGE_NON_LINEAR;



  DCL ATOMIC ENTRY(CHAR(25)VAR,FIXED BINARY);
     ATOMIC:PROCEDURE(A,ND); DCL A CHAR(*),ND FIXED BIN;
 DCL STRING CHAR(2000)VAR,C CHAR(20) VAR,IB FIXED BIN;
 IB=INDEX(A,'(')-1; C=SUBSTR(A,1,IB)||'1'; OPEN FILE(SYSCP)TITLE(C)
   OUTPUT;
 DCL NT(4); NT=1;DO NJ=1 TO ND;NT(NJ)=MODES;END;
 ALL:DO I=1 TO NT(1);DO J=1 TO NT(2); CET(I;J);
 DO K=1 TO NT(3); CET(K); DO L=1 TO NT(4);CET(L);
 IF ¬DENFMCG('Z9999997='||A,'Z9999998=0') THEN DO;
 CALL DENFMCH(STRING,A); CALL DENFMC8(A||'=0');
 PUT FILE(SYSCP)LIST(STRING)SKIP;
 END;
 END ALL;
 CLOSE FILE(SYSCP);
 RETURN;
 END ATOMIC;




1
 INPUT:PROCEDURE;
 /*1.*/ GET FILE(CONTROL)DATA (MODES,
 SECOND_ORDER,
         NOTHING);
 MODAL=MODES/2; IF 2*MODAL=MODES THEN NOD=1; ELSE NOD=0;
 TOTAL=0; NEW_TOTAL=0;
 OPTSET(EXPND);

 /*2.*/ /* ZERO THE ARRAYS OF COEFFICIENTS */
         /*A*/ CALL FORMAC_ERASE('D(I)',MODES,1);
         /*B*/ CALL FORMAC_ERASE('S1(I,J)',MODES,2);
         /*C*/ CALL FORMAC_ERASE('GAMMA(I,J,K)',MODES,3);
         /*D*/ CALL FORMAC_ERASE('S(I,J)',MODES,2);
         /*E*/ CALL FORMAC_ERASE('M(I,J)',MODES,2) ;

 FORMAC_ERASE:PROCEDURE(A,N,RANK);
 DCL A CHAR(*),(N,RANK,NT(4) ) FIXED BINARY;
```

```
 DO I=1 TO 4; IF I<=RANK THEN NT(I)=N; ELSE NT(I)=1; END;
ZERO: DO I=1 TO NT(1);CET(I); DO J=1 TO NT(2);CET(J);
DO K=1 TO NT(3); CET(K); DO L=1 TO NT(4); CET(L);
RET(A=0); END     ZERO; ATOMIZE(I;J;K;L); END FORMAC_ERASE;

/*3.*/  /* INPUT THE ARRAYS*/

        /*A*/ CALL EQUATIONS('DRIVER');
        /*B*/ CALL EQUATIONS('S');
        /*C*/ CALL EQUATIONS('GAMMAS');

EQUATIONS:PROCEDURE(A); DCL A CHAR(*);
OPEN FILE(SYSIN) TITLE(A) INPUT;
ON ENDFILE(SYSIN) BEGIN; CLOSE FILE(SYSIN); GOTO EOF; END;
LOOP: GET FILE(SYSIN) LIST(VALUE)COPY; FORM(VALUE); GOTO LOOP;
EOF: RETURN; END EQUATIONS;

DO IB=1 TO MODES;CET(IB); LET(MU(IB)=EXPAND(-#I*MU(IB))); END;
LET(MULIST=(MU.(1),MU(1)));
DO IS=2 TO MODES; CET(IS);
     LET(MULIST=(MULIST,MU.(IS),MU(IS)    )); END;
PRINT_OUT(MULIST);

/*4.*/ /* FORM THE SECOND ORDER EQUATIONS FOR REFERENCE */
DCL       SECOND_ORDER BIT(1) INIT('1'B);
DCL SECOND_ORDER_EQUATIONS ENTRY;   IF SECOND_ORDER THEN CALL
               SECOND_ORDER_EQUATIONS;
ELSE GOTO EOSH;
 SECOND_ORDER_EQUATIONS:PROC;  DO I=1 TO MODES; CET(I);
LET(DRIVE(I)=DERIV(D(I),TIME));
  DO IS=1 TO MODES; CET(IS);LET(DRIVE(I)=DRIVE(I)+M(I, IS )*D( IS ));
  DO J=1 TO MODES; CET(J);    LET(LINEAR(I,J)=0 ); DO IS=1 TO MODES; CET
(IS);LET(LINEAR(I,J)=LINEAR(I,J)+M(I, IS )*M( IS ,J));END;
  DO K=1 TO MODES; CET(K);  LET(MATHIEU(I,J,K)=GAMMA(I,J,K)+GAMMA(I,K,J
));
     LET(QUAD(I,J,K)=0 ); DO IS=1 TO MODES; CET(IS);LET(QUAD(I,J,K)=
QUAD(I,J,K)+M(I, IS )*GAMMA( IS ,J,K)+M( IS ,J)*GAMMA(I, IS ,K)    +M(
IS ,K)*GAMMA(I,J, IS ));END;
  DO L=1 TO MODES; CET(L);     LET(TRIPLE(I,J,K,L)=0 ); DO IS=1 TO MODES
; CET(IS);LET(TRIPLE(I,J,K,L)=TRIPLE(I,J,K,L)+GAMMA(I, IS ,K)*GAMMA( IS
,J,L)+GAMMA(I,J,IS)*GAMMA(IS,K,L)); END;
END;    END;   END;
END SECOND_ORDER_EQUATIONS;


/*5.*/ /* PRINT OUT THE TERMS OF SECOND ORDER */
CALL FORMAC_PRINT('DRIVE(I)',1);
CALL FORMAC_PRINT('LINEAR(I,J)',2);
CALL FORMAC_PRINT('QUAD(I,J,K)',3) ;
CALL FORMAC_PRINT('MATHIEU(I,J,K)',3);
CALL FORMAC_PRINT('TRIPLE(I,J,K,L)',4);


FORMAC_PRINT:PROC(A,RANK); DCL A CHAR(*),(RANK,NT(4))FIXED BIN;
PUT PAGE;
NT=1; DO I=1 TO RANK; NT(I)=MODES;END;
PRINT_ALL:
DO I=1 TO NT(1);CET(I); DO J=1 TO NT(2); CET(J);
DO K=1 TO NT(3); CET(K); DO L=1 TO NT(4); CET(L);
IF ¬DENFMCG('Z9999997='||A,'Z9999998=0' ) THEN
CALL DENFMC2(A); CALL DENFMC8(A||'=0');
END; END; END; END PRINT_ALL;
END FORMAC_PRINT;

DCL FORMAC_PRINT ENTRY(CHAR(20)VAR,FIXED BINARY) ;
DCL FORMAC_ERASE ENTRY(CHAR(15)VAR,FIXED BIN,FIXED BIN),
```

```
    EQUATIONS ENTRY (CHAR(6) VAR);


 GOTO EOSR;  OUTPUT:ENTRY;
 PUT FILE(PUNCH)LIST(STRING) SKIP;

 EOSR: END INPUT;








ITERATE: PROCEDURE;
FIRST_ORDER_ITERATION:ENTRY; ITERATE#=1;
ITERATIONS: DO ITER=1 TO ITERATE#; CET(KAPPA=ITER);  CET(ITER);
LET(KAPPA1=KAPPA-1);    KAPPA1=ITER-1;
MODE: DO I=1 TO MODES; CET(I);
/*1.*/ CALL SUBSTITUTE;
/*2.*/ CALL INTEGRATE;
END MODE;
END ITERATIONS;
EOSR: RETURN; END ITERATE;


SUBSTITUTE:PROCEDURE;
LET(I$=0);
JMODES: DO J=1 TO MODES; CET(J);
KMODES: DO K=1 TC MODES; CET(K);
        IF IDENT(GAMMA(I,J,K);0) THEN GOTO K_END;
        LOWER_ORDERS: DO SIGMA=0 TO KAPPA1; KSIG=KAPPA1-SIGMA;
            CET(SIGMA;KSIG);
            JTERMS: DO JN=1 TO TOTAL(SIGMA ,J); CET(JN);
             KTERMS: DO KN=1 TO TOTAL(KSIG  ,K); CET(KN);
            LET(I$=I$+1;
             NEWTERM(I$)=GAMMA(I,J,K)*B(SIGMA,J,JN)*B(KSIG,K,KN);
             EXPONENT(I$)=EXPONENT(SIGMA,J,JN)+EXPONENT(KSIG,K,KN);
             );

SAVE( EXPONENT(I$);EXPONENT(SIGMA,J,JN);EXPONENT(KSIG,K,KN);
NEWTERM(I$);B(SIGMA,J,JN);B(KSIG,K,KN)    );

            END KTERMS;
            END JTERMS;
        END LOWER_ORDERS;
K_END:
END KMODES;
END JMODES;
NEW_TOTAL(I)=INTEGER(I$);
 EOSR: RETURN; END SUBSTITUTE;




INTEGRATE:PROCEDURE;
LET(EIGEN=MU.(I)*#1; J$=-1);
IMODES:DO I$=1 TO NEW_TOTAL(I); CET(I$);
LET(EXPONA=EXPONENT(I$)-EIGEN);
IF ¬IDENT(EXPONA:0) THEN DO;
 LET(J$=J$+2;
         EXPONENT(ITER,I,J$)=EXPONENT(I$);
         EXPONENT(ITER,I,J$+1)=EIGEN;
    B(ITER,I,J$)=NEWTERM(I$)/EXPONA; B(ITER,I,J$+1)=-B(ITER,I,J$) );
    IF IDENT(B(ITER,I,J$);0) THEN LET(J$=J$-2);
ATOMIZE(NEWTERM(I$);EXPONA;TEEFOW(I$);EXPONENT(I$););
END;
```

```
ELSE PUT LIST(I,I$,'SECULARITY');
END IMODES;
TOTAL(ITER,I)=INTEGER(J$)+1;
DO I#=1 TO TOTAL(ITER,I) ; CET(I#); SAVE(
B(ITER,I,I#);TEEPOW(ITER,I,I#);EXPONENT(ITER,I,I#) ) ;      END;
EOSR: RETURN;
END INTEGRATE;


SAVER:PROC; DO ORDER=0 TO 1; CET(ORDER);
 DO J=1 TO MODES; CET(J) ;
DO K$=1 TO TOTAL(J,ORDER);
CET(K$);
PRINT_OUT(B(ORDER,J,K$));
CHARFX(STRING=B(ORDER,J,K$)); PUT FILE(SAVED)LIST(STRING); END  SAVER;

DCL (FORM_NEW_DRIVER,CHANGE_NON_LINEAR,LOWEST_ORDER_SOLUTION,
FIRST_ORDER_ITERATION,SUBSTITUTE,INTEGRATE,SAVER,
EVALUATE_NON_LINEAR_TERMS,EIGENVALUE_CORRECTION) ENTRY;
DCL TRANSMOGRIFY ENTRY (CHAR(20)VAR);
DECLARE
        VALUE CHAR (800) VAR,
        NEW_TOTAL(100) FIXED BINARY,
        (SIGMA,ORDER)FIXED BIN,
        OUTPUT ENTRY,
        INPUT ENTRY,
        TOTAL(40,20) FIXED BIN,
        STRING CHAR(10000) VARYING,
        DENFMC3 ENTRY (FIXED BIN(31,0),FIXED BIN(31,0)),
        NOTHING FIXED;
END_OF_PROGRAM: END THIRD;
// EXEC EXPORT,PROGRAM=EDITY
//STEPLIB DD DSN=U.FPL.LIBRARY,DISP=SHR
//SYSPRINT DD SYSOUT=A
//IN DD DSN=U.ROSEN.ARPA(THREE),DISP=SHR,UNIT=SYSDA,VOL=SER=RES103
//OUT DD SYSOUT=A
//SYSIN DD *
```

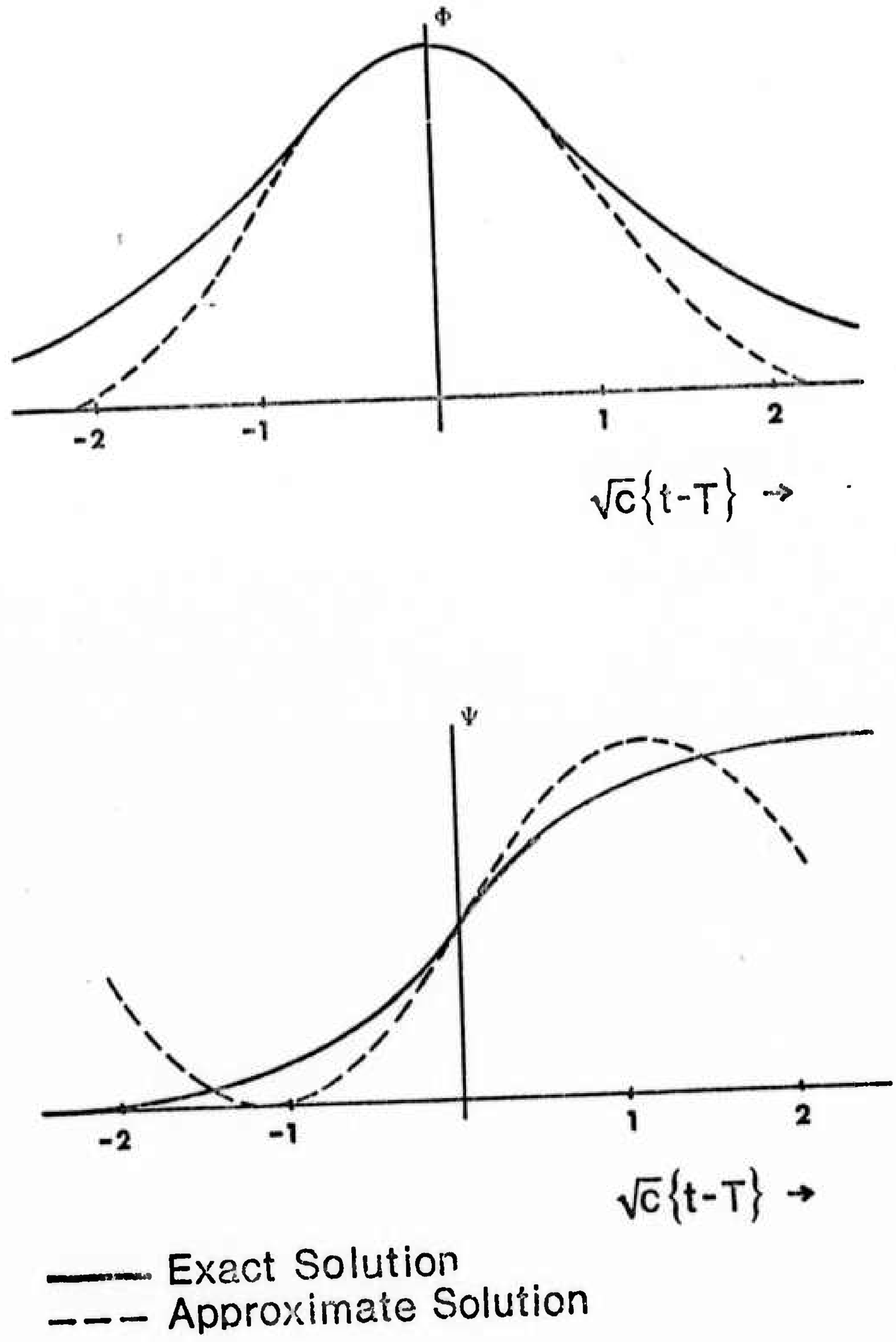
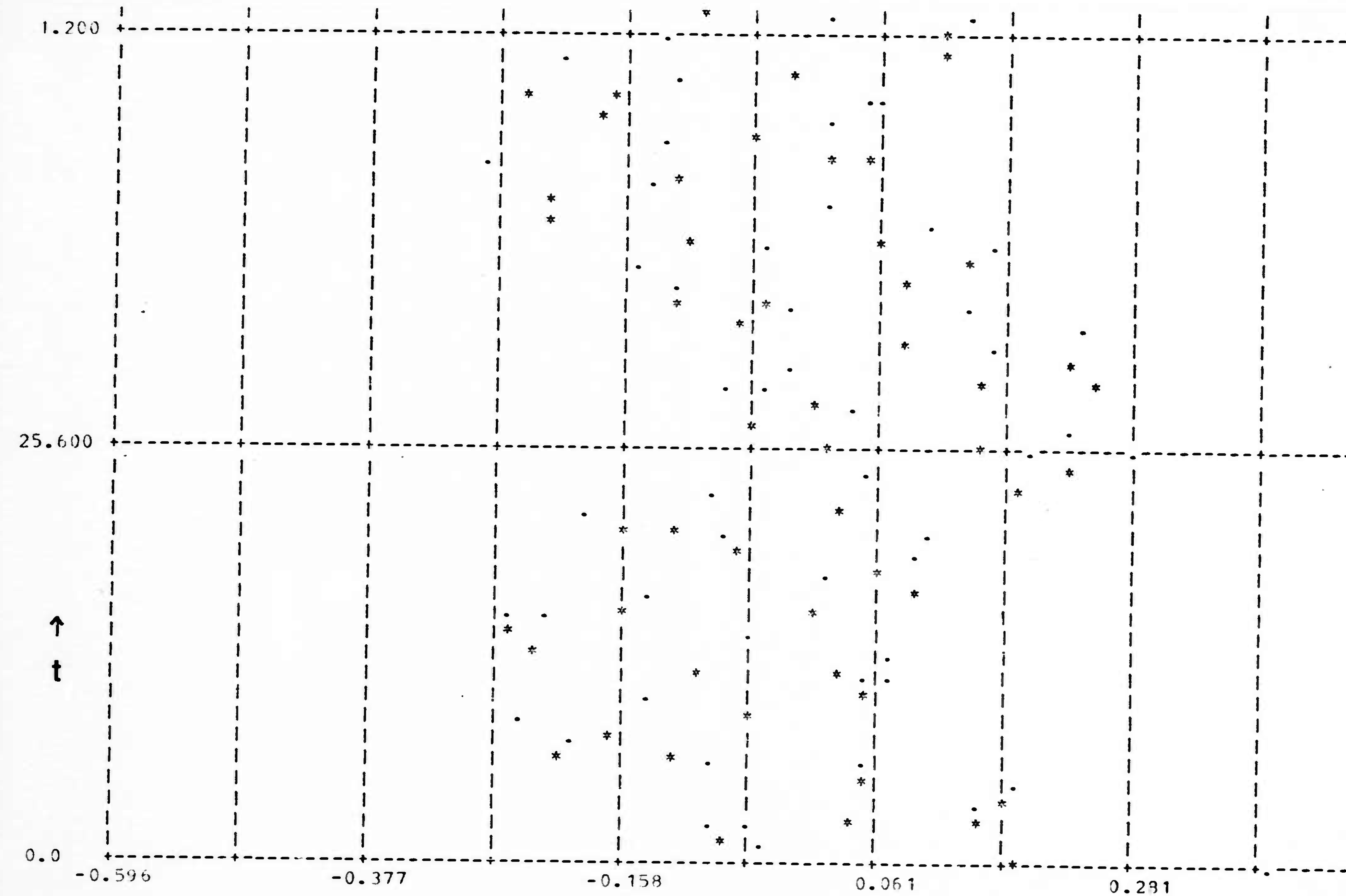


Figure 1

```
) **(-1 ) *(1/4 )-S1(2,2) *S1(1,3) *S(3,1) *S(1,3) *S(2,1) *KA**2*MU.(2) *MU.(3) *(WO*MU.(2) *MU.(3) **2*2 + WO*MU.(2) **2*MU.(3) *2
+ WO**2*MU.(2) *MU.(3) *4 + WO**2*MU.(2) **2 + WO**2*MU.(3) **2 + WO**3*MU.(2) *2 + WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 +
WO**4) **(-1 ) *(1/4 )
SUMMA(1,21) = -S1(3,2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *WO*MU.(2) *(-WO*MU.(2) *MU.(3) **2*2-WO*MU.(2) **2*MU.(3) *2 + WO**2*MU
.(2) *MU.(3) *4 + WO**2*MU.(2) **2 + WO**2*MU.(3) **2-WO**3*MU.(2) *2-WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(
1/4 )-S1(3,2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *WO*MU.(3) *(-WO*MU.(2) *MU.(3) **2*2-WO*MU.(2) **2*MU.(3) *2 + WO**2*MU.(2) *MU.(3
) *4 + WO**2*MU.(2) **2 + WO**2*MU.(3) **2-WO**3*MU.(2) *2-WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(1/4 ) + S1
(3,2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *MU.(2) *MU.(3) *(-WO*MU.(2) *MU.(3) **2*2-WO*MU.(2) **2*MU.(3) *2 + WO**2*MU.(2) *MU.(3) *4
+ WO**2*MU.(2) **2 + WO**2*MU.(3) **2-WO**3*MU.(2) *2-WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(1/4 ) + S1(3,
2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *WO**2*(-WO*MU.(2) *MU.(3) **2*2-WO*MU.(2) **2*MU.(3) *2 + WO**2*MU.(2) *MU.(3) *4 + WO**2*MU
.(2) **2 + WO**2*MU.(3) **2-WO**3*MU.(2) *2-WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(1/4 )
SUMMA(1,22) = S1(3,2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *WO*MU.(2) *(WO*MU.(2) *MU.(3) **2*2-WO*MU.(2) **2*MU.(3) *2-WO**2*MU.(2) *
MU.(3) *4 + WO**2*MU.(2) **2 + WO**2*MU.(3) **2 + WO**3*MU.(2) *2-WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(1/4
)-S1(3,2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *WO*MU.(3) *(WO*MU.(2) *MU.(3) **2*2-WO*MU.(2) **2*MU.(3) *2-WO**2*MU.(2) *MU.(3) *4 +
WO**2*MU.(2) **2 + WO**2*MU.(3) **2 + WO**3*MU.(2) *2-WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(1/4 )-S1(3,2) *
S1(2,3) *S(3,1) *S(1,1) *S(2,1) *MU.(2) *MU.(3) *(WO*MU.(2) *MU.(3) **2*2-WO*MU.(2) **2*MU.(3) *2-WO**2*MU.(2) *MU.(3) *4 + WO**2*
MU.(2) **2 + WO**2*MU.(3) **2 + WO**3*MU.(2) *2-WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(1/4 ) + S1(3,2) *S1(2
,3) *S(3,1) *S(1,1) *S(2,1) *WO**2*(WO*MU.(2) *MU.(3) **2*2-WO*MU.(2) **2*MU.(3) *2-WO**2*MU.(2) *MU.(3) *4 + WO**2*MU.(2) **2 +
WO**2*MU.(3) **2 + WO**3*MU.(2) *2-WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(1/4 )
SUMMA(1,23) = -S1(3,2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *WO*MU.(2) *(-WO*MU.(2) *MU.(3) **2*2 + WO*MU.(2) **2*MU.(3) *2-WO**2*MU
.(2) *MU.(3) *4 + WO**2*MU.(2) **2 + WO**2*MU.(3) **2-WO**3*MU.(2) *2 + WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 )
*(1/4 ) + S1(3,2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *WO*MU.(3) *(-WO*MU.(2) *MU.(3) **2*2 + WO*MU.(2) **2*MU.(3) *2-WO**2*MU.(2) *
MU.(3) *4 + WO**2*MU.(2) **2 + WO**2*MU.(3) **2-WO**3*MU.(2) *2 + WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(1/4
)-S1(3,2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *MU.(2) *MU.(3) *(-WO*MU.(2) *MU.(3) **2*2 + WO*MU.(2) **2*MU.(3) *2-WO**2*MU.(2) *MU.(
3) *4 + WO**2*MU.(2) **2 + WO**2*MU.(3) **2-WO**3*MU.(2) *2 + WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(1/4 )
+ S1(3,2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *WO**2*(-WO*MU.(2) *MU.(3) **2*2 + WO*MU.(2) **2*MU.(3) *2-WO**2*MU.(2) *MU.(3) *4 +
WO**2*MU.(2) **2 + WO**2*MU.(3) **2-WO**3*MU.(2) *2 + WO**3*MU.(3) *2 + MU.(2) **2*MU.(3) **2 + WO**4) **(-1 ) *(1/4 )
SUMMA(1,24) = S1(3,2) *S1(2,3) *S(3,1) *S(1,1) *S(2,1) *WO*MU.(2) *(WO*MU.(2) *MU.(3) **2*2 + WO*MU.(2) **2*MU.(3) *2 + WO**2*MU
```

Figure 2



* -approximate

· -exact

A →

Figure 3

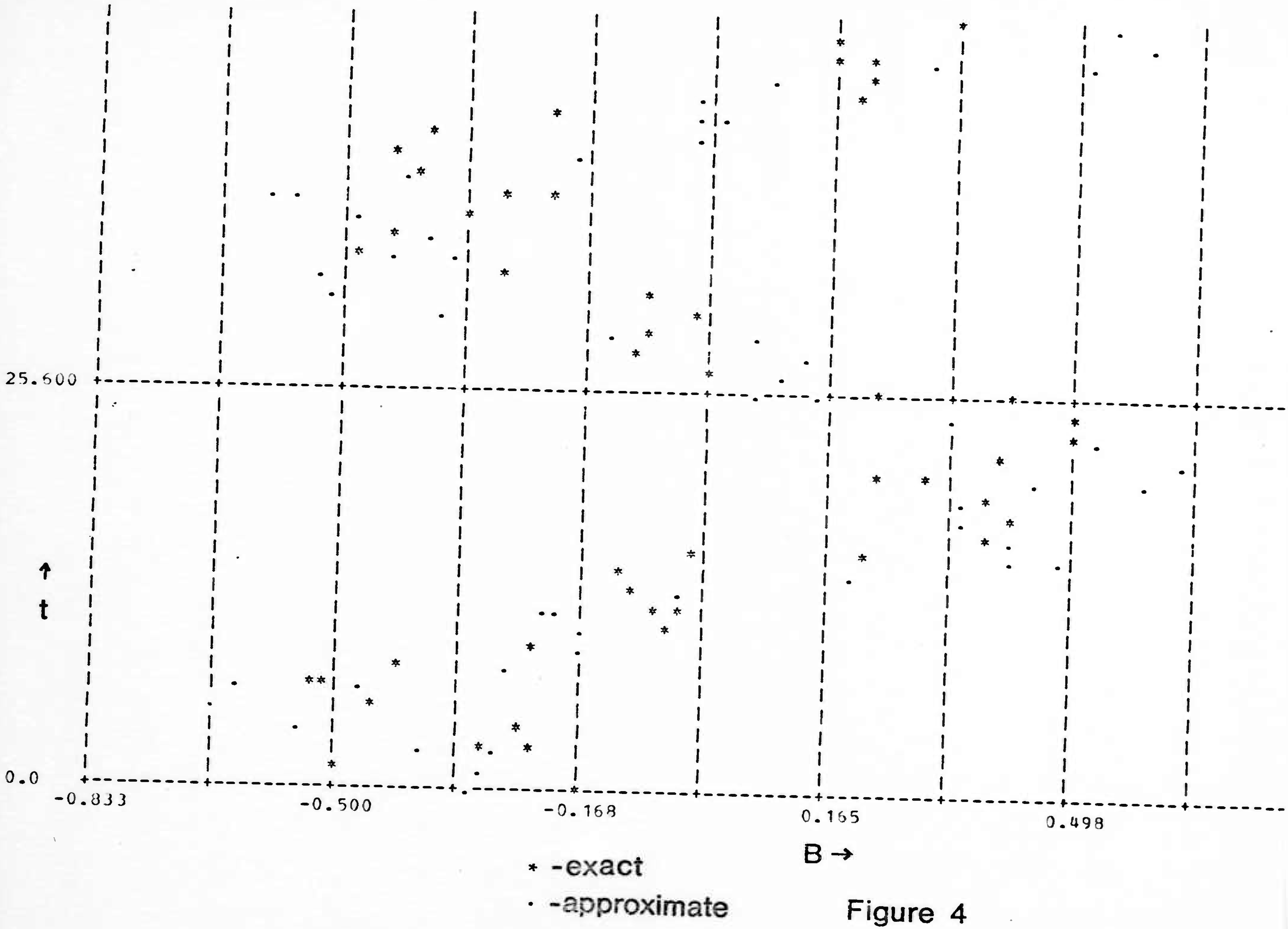


Figure 4

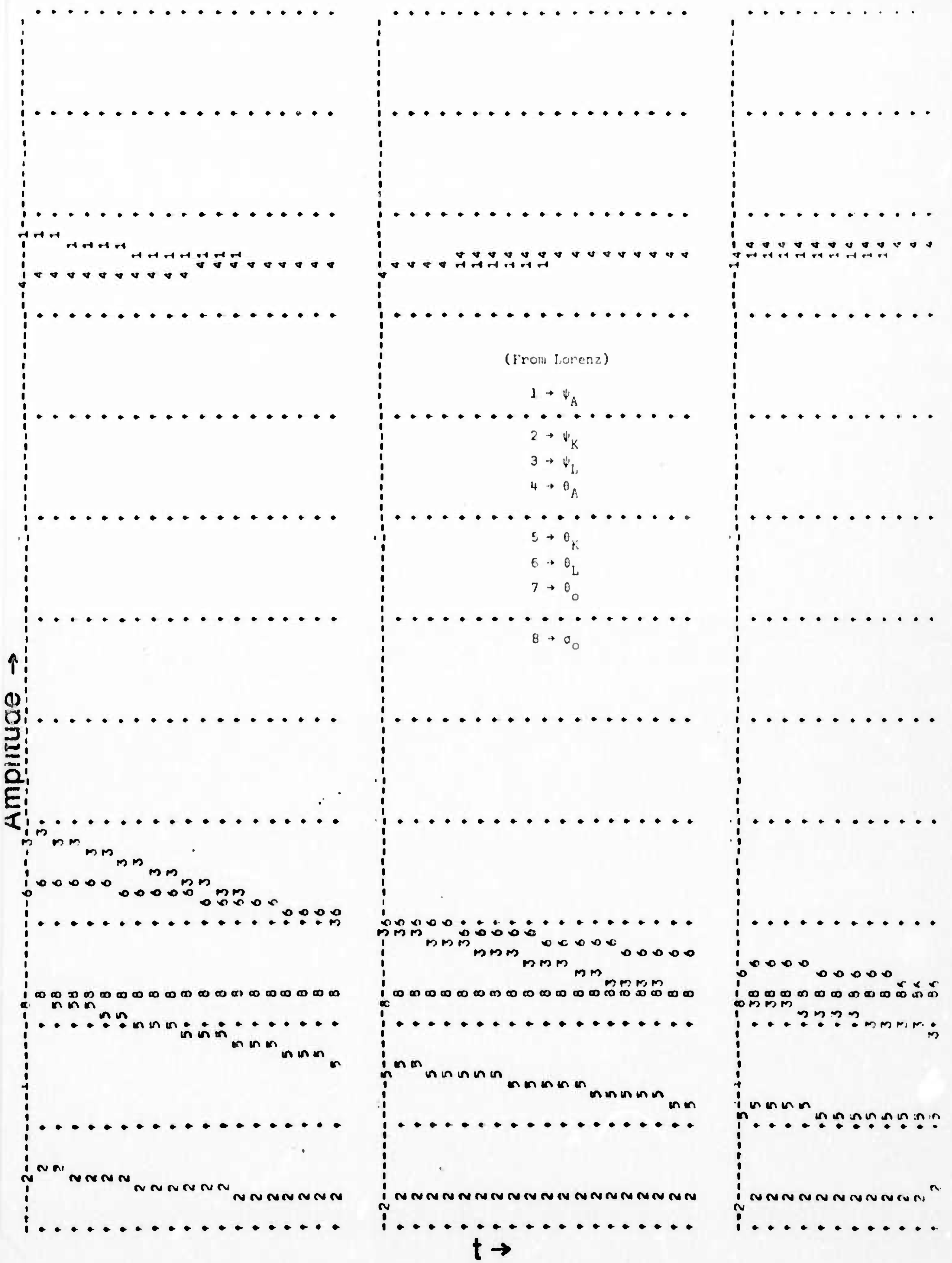

Amplitude →
(From Lorenz)
1 → ψ_A
2 → ψ_K
3 → ψ_L
4 → θ_A
5 → θ_K
6 → θ_L
7 → θ_o
8 → σ_o
t →

Figure 5

<u>References</u>

1. R. Tobey et. al., PL/1 FORMAC Interpreter, SHARE contributed Library, 360D03.3.004

   J. Sammet, "Formula Manipulation by Computer", Adv. in Computers, Vol. 8, (Eds. F. AH and M. Rubinoff), Academic Press, New York, 1967.

2. MACSYMA Reference Manual, MATHLAB Group, Project MAC, M.I.T., Cambridge, Mass., Version Five, June 1973.

3. N. Minorsky, "Non Linear Oscillations", D. Van Nostrand, Princeton, 1962.

   N. N. Bogoliubov and Y. A. Mitropolsky, "Asymptotic Methods in the Theory of Non-Linear Oscillations", (English Translation), Gordon and Breach, New York, 1961.

4. H. T. Davis, "Introduction to Nonlinear Differential and Integral Equations", Dover Publications, New York 1962.

5. Barry Saltzman and Chung-Muh Tang, Analytical Study of the Evolution of an Amplifying Baroclinic Wave, J. Atmos. Sci., 29, 427-444, 1972.

6. E. N. Lorenz, Modified Dynamic Equations Applied to the Rotating Basin Experiment, J. Atmos. Sci., 19, 39-51, 1962.